

REMARKS

In view of the foregoing amendments and following remarks responsive to the Office Action dated April 27, 2006, Applicant respectfully requests favorable reconsideration of this application.

Responses to Objections and Rejections as to Claim Form

Hyperlink

In section 12 of the Office Action, the Office objected to the specification because it contains "an embedded hyperlink and/or other form of browser-executable code". In response to this same objection contained in the previous Office Action, Applicant amended the specification so as to make the hyperlink not browser-executable. In section 3 of the Response to Arguments section of the present Office Action, the Office indicated that "Adding spaces to the text of the hyperlink to allegedly make it not browser-executable does not solve this issue. The hyperlink should be removed from the specification".

Applicant respectfully traverses. MPEP section 608.01, explains:

Examiners must review patent applications to make certain that hyperlinks and other forms of browser-executable code, especially commercial site URLs, are not included in a patent application. >37 CFR 1.57(d) states that an incorporation by reference by hyperlink or other form of browser executable code is not permitted.< Examples of a hyperlink or a browser-executable code are a URL placed between these symbols "< >" and http:// followed by a URL address. When a patent application with embedded hyperlinks and/or other forms of browser-executable code issues as a patent (or is published as a patent application publication) and the patent document is placed on the USPTO web page, when the patent document is retrieved and viewed via a web browser, the URL is interpreted as a valid HTML code and it becomes a live web link. When a user clicks on the link with a mouse, the user will be transferred to another web page identified by the URL, if it exists, which could be a commercial web site. USPTO policy does not permit the USPTO to link to any commercial sites since the USPTO exercises no control over the organization, views or accuracy of the information contained on these outside sites.

Accordingly, the issue with respect to hyperlinks in specifications is, in fact, whether or not they are browser-executable. The very statement in the Office Action that "the disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code" itself clearly indicates that the hyperlink must be browser executable in order to be objectionable. Specifically, it expressly uses the term "other browser-executable code", thus explicitly indicating that the term "embedded hyperlink" in the statement is, by definition, browser-executable code. Accordingly, adding spaces to make the phrase not browser-executable, by definition, makes it not objectionable. Applicant has herein further amended this paragraph to eliminate "http://" from the term, thus further assuring that it cannot be executed by a browser. Applicant respectfully requests Office to withdraw this objection.

Non-Enablement Rejection

In section 14 of the Office Action, the Office rejected claims 1-25 under 35 U.S.C. 112, first paragraph as non enabled. This rejection also is held over from the previous Office Action. The Office asserts that the claims recite a computer program product, but also recite that the computer program product is in the form of a Web service and that a Web service is a method that can be performed over the World Wide Web. The Office also asserts that a computer program that recorded on a computer readable medium is, by definition, not a method, since it is tangibly embodied".

Applicant respectfully traverses. Specifically, as the Office expressly stated in section 5 of the present Office Action:

In order to accurately analyze Applicants arguments, Applicants specification must be considered. On page 3 of the original specification, Applicant stated:

Web services is a term applied to application logic or application software modules that can be exposed to and shared with others over the Internet via a standardized interface mechanism.

Furthermore, in sections 6 and 7 of the present Office Action, the Office essentially admits that Applicant's definition of "Web services" is consistent with the Office's interpretation of "Web services".

Accordingly, by the Office's own admission, a Web service is an "application software module". Certainly, the Office cannot be disputing that an application software module can be a "a computer program product" or that a computer program product can be recorded on computer readable medium. As such, Applicant respectfully requests the Office to withdraw this rejection.

Indefiniteness Rejections

In section 17 of the Office Action, the Office rejected claims 1, 12, 15-19, 23, 25, and 38 as indefinite under 35 U.S.C. 112, second paragraph, because it is unclear what Applicant means by "causing the dynamic reconfiguration of Web services based on said messages and said Web services available at said corresponding network node". The Office asserted that the language is confusing because it is unclear (a) if the messages are at the corresponding network node, (b) the Web services are active at the corresponding network node, (c) the Web service can be downloaded at the corresponding network node, or (d) if the messages are local. Further, the Office asserted that (e) no standard was given in the specification for the dynamic reconfiguration of a Web service. This rejection also is held over from the previous Office Action.

Applicant respectfully traverses. As noted in the specification at, for instance, page 6, first sentence, and the numerous reconfiguration examples provided on pages 19-39, the present invention provides a software construct for managing Web services at a network node and an adaptive model for the dynamic reconfiguration of the plurality of Web service containers distributed throughout the network. This is accomplished using the messages sent back and forth across the network by the various containers communicating with each other. Thus, in response to the Office's five specific points, with regard to (b) and (c) above (i.e., whether the Web services are active or can be downloaded at the corresponding network node), the claim intentionally

does not specify whether the services are active at the corresponding network node or can be downloaded at the corresponding network because that is what is being dynamically reconfigured in the invention. The entire invention concerns how containers change whether a Web service is active at a particular node or downloadable. It would be contrary to the very invention that Applicant is trying to claim if the independent claim specified whether a particular Web service is active or available for download at a particular node.

Furthermore, with respect to items (a) and (d) above, i.e., whether the messages are local or at the corresponding network node, the claim clearly recites "causing the dynamic reconfiguration of said Web services available at said corresponding network node on said network based on said transmitted and received messages". The claim also recites "generating messages to be transmitted to other containers" and "receiving and deciphering messages disclosing Web services that are available at other network nodes corresponding to other containers". Accordingly, the claim very clearly recites that the messages comprise both messages received from other nodes and messages generated at the corresponding node. Thus, in specific response to the Office's inquiry, the "transmitted messages" are sent from the corresponding node to other nodes and the "received messages" are received at the corresponding node from other nodes. This is exactly what the claim recites.

If this explanation and the response to this same rejection that was contained in the previous response filed by Applicant does not satisfy the Office, Applicant respectfully requests the Office to provide further explanation of this rejection, as Applicant does not understand what the Office finds objectionable about this claim or what the Office is looking for in response to this rejection.

With respect to item (e) above, i.e., the "standard" of dynamic reconfiguration of a Web service, Applicant is not certain what the Office is looking for with respect to this matter. Applicant respectfully traverses as the specification is replete with discussion of what is meant by dynamically reconfiguring Web services. Furthermore, the terminology itself is sufficiently clear. Specifically, Applicant and the Office have agreed

on what is meant by Web services (see discussion above). Thus, this leaves the terms "dynamically" and "reconfigurable" for interpretation. However, Applicant is using these terms in their ordinary and normal meaning. For instance, according to The American Heritage dictionary, Second College Edition, "dynamic" means "characterized by continuous change, activity, or progress". This is exactly what Applicant means by "dynamically". This dictionary also defines "configure" as "to give configuration to; form; shape". Again, by "reconfigure", Applicant means changing the configuration of [Web services on the network]. The specification of the present application is replete with examples of reconfiguring of Web services, including, for instance: "businesses may configure their containers to deliver Web service module code only when the request is from a container with the proper security credentials and/or is located behind the corporate firewall" (page 6, lines 12-14); "send and receive Web service software modules to and from other Web service containers" (page 6, lines 14-16); "exchange Web services software as well as contextual information" (page 7 lines 1-2); "reconfigure themselves as routers" (page 7, line 4); "load and unload Web service software modules" (page 7, lines 4-5); "send software modules to peer containers" (page 7, line 6); "acting as a 'service router'" (page 7, last line through page 8, line 5); and all of the specific reconfiguration scenarios described on pages 19-39 of the specification.

Nevertheless, in an attempt to move the prosecution of this application forward, has amended independent claims 1 and 25 to now specifically recite that reconfiguring includes at least exchanging Web services software modules between network nodes, such as specifically discussed at page 6, lines 12-16 and page 7, line 6 of the specification, for example.

Accordingly, Applicant respectfully requests the Office to withdraw this rejection.

The Office further rejected claim 13 under 35 U.S.C. 112, second paragraph, asserting that the claim is unclear whether the response being received to said request is a message indicating the Web service can be downloaded or the downloaded Web service software. The Office asserted that Applicant's arguments do not reflect the

claim language. The Office asserted that "if Applicant intended the response of claim 13 to be strictly a list of the available services, and not the downloaded code, Applicant should amend the claim appropriately".

Applicant has amended the claims to more clearly recite the feature claimed in claim 13 as well as many other claims. In drafting the claims for this application, Applicant intended to use the term "client request" to refer to requests from client machines that seek to use a Web service and the term "message" to refer to messages exchanged between Web service containers concerning Web services. Such messages, of course, might comprise requests for copies of Web services, requests for information about Web services at other containers, etc.

Unfortunately, the former claim language contained unintended discrepancies in terminology. Specifically, the former claims inadvertently used the terms "request" and "client request" interchangeably. Also, in two instances, the claims used the term "request" to refer to a "message".

Applicant has herein amended the claims to eliminate these discrepancies.

Thus referring now to amended claim 13, it specifically recites "computer executable instructions for routing said client requests for a Web service that is not available at said corresponding network node and has been determined to be available at another network node to another container corresponding to said other network node" and "computer executable instructions for receiving responses to said client requests from said another network node". The preamble of claim 13 recites that the body of the claim describes the "proxy" recited in claim 12. Claim 12 recites that the proxy is invoked "responsive to receipt of one of said client requests from a client for a Web service that is not available at said corresponding network node". Accordingly, the "response" in claim 13 is a response to a client request for use of a particular Web service that has been sent from a client machine to the container corresponding to another node on the network and that was forwarded from that other node to the claimed container. Accordingly, it should now be clear that the "response" recited in claim 13 is neither a list of available services or a copy of an available service. Rather,

it is a response to a client request for use of a particular Web service. Thus, using the example discussed on pages 20-21 of the specification, it the response that would contain the value of the converted currency.

Responses to Prior Art Rejections

The Office has essentially maintained its prior art rejections also. Specifically, the Office rejected claims 1-4, 7-9, 11-14, 22-23, 25-28, 31-33, 35-38, and 46-47 as anticipated by Zintel. The Office further rejected claims 5 and 29 as obvious over Zintel in view of Christensen, claims 6 and 30 as obvious over Zintel in view of Christensen and further in view of Januszewski; claims 10 and 34 as obvious over Zintel in view of project JXTA, claims 15-21 and 39-45 as obvious over Zintel in view of Jindal, and claims 24 and 48 as obvious over Zintel.

Since the prior art rejections are essentially the same as asserted in the previous Office Action and since Applicant has already specifically addressed those rejections, Applicant will herein focus on the Office's comments in the Response to Arguments section of the final Office Action.

It is clear from those responses that the primary issue in dispute is what is meant by the term "Web service". Specifically, the Office asserted that Applicant's definition of Web service is found on page 3 of the specification, namely:

Web services is a term applied to application logic or application software modules that can be exposed to and shared with others over the Internet via a standardized interface mechanism.

The Office asserted that:

Applicant's express definition of Web services is equivalent to Zintel's definition of a service in Column 9, lines 1-12, which disclosed processes such as a clock available on devices in Zintel which were accessed over the Web (column 1, line 35). The capabilities of the device (column 2, lines 1-4) were Web services since they were performed over a network or the Web. See further column 4, lines 57-67. Furthermore Applicants "express definition" of Web services could be legitimately read as any software used in connection with a network.

Applicant respectfully traverses. The Office appears to be selectively picking and choosing portions of the specification to serve its definition of Web services as reading on any software on a network, but ignoring parts of the specification that clearly refine the definition of "web services". For instance, the three sentences following the sentence quoted above by the Office state:

The standard paradigm on the Web is based on the exchange of files containing displayable information, e.g., web pages. Thus, the Web services concept can be considered an extension of this paradigm to automated exchange of software modules between nodes of the network, i.e., machine-to-machine, or business-to-business interfaces. Furthermore, the receiving node can automatically set up and run the software without human intervention.

Thus, Applicant's express definition of Web services cannot legitimately be read on any software used on a network, but is limited to software modules that can be exchanged between nodes of a network and that can be run at the receiving node. Applicant has amended the independent claims to now recite these limitations.

In section 8 of the Response to Arguments section of the Office Action, the Office notes that MPEP 2111.01 recites that:

While the claims of issued patents are interpreted in light of the specification, prosecution history, prior art and other claims, this is not to the mode of claim interpretation to be applied during examination. During examination, the claims must be interpreted as broadly as their terms reasonably a lateral.

Once again, the Office is picking and choosing portions and statements in the MPEP selectively so as to defend interpretations which are clearly not warranted in view of the entire document. Specifically, the above quoted portion of MP EP 2111.01 is followed immediately by the following sentence

This means that the words of the claim must be given their plain meaning unless Applicant has provided a clear definition in the specification.

Applicant contends that it is using the term Web services in its ordinary and usual meaning. The W3C defines a Web service as follows:

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services are characterized by their great interoperability and

extensibility, as well as their machine-processable descriptions thanks to the use of XML. They can be combined in a loosely coupled way in order to achieve complex operations. Programs providing simple services can interact with each other in order to deliver sophisticated added-value services.
(<http://www.w3.org/2002/ws/Activity>).

Wikipedia defines Web services as follows:

According to the W3C a **Web service** is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface that is described in a machine-processable format such as WSDL. Other systems interact with the Web service in a manner prescribed by its interface using messages, which may be enclosed in a SOAP envelope, or follow a RESTful approach. These messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other Web-related standards. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (for example, between Java and Python, or Microsoft Windows and Linux applications) is due to the use of open standards. DASIS and the W3C are the primary committees responsible for the architecture and standardization of web services. To improve interoperability between web service implementations, the WS-I organization has been developing a series of profiles to further define the standards involved. (<http://en.wikipedia.org/wiki/Web%5Fservice>)

Webopedia defines Web services as follows:

The term *Web services* describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. The applications interface, not the users. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

Web services allow different applications from different sources to communicate with each other without time-consuming custom coding, and because all communication is in XML, Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl, Windows applications can talk with UNIX applications.

Web services do not require the use of browsers or HTML.
(http://www.webopedia.com/TERM/W/Web_services.html)

Whatis.com defines Web services as follows:

Web services (sometimes called *application services*) are services (usually including some combination of programming and data, but possibly including human resources as well) that are made available from a business's Web server for Web users or other Web-connected programs. Providers of Web services are generally known as application service providers. Web services range from such major services as storage management and customer relationship management (CRM) down to much more limited services such as the furnishing of a stock quote and the checking of bids for an auction item. The accelerating creation and availability of these services is a major Web trend.

Users can access some Web services through a peer-to-peer arrangement rather than by going to a central server. Some services can communicate with other services and this exchange of procedures and data is generally enabled by a class of software known as middleware. Services previously possible only with the older standardized service known as Electronic Data Interchange (EDI) increasingly are likely to become Web services. Besides the standardization and wide availability to users and businesses of the Internet itself, Web services are also increasingly enabled by the use of the Extensible Markup Language (XML) as a means of standardizing data formats and exchanging data. XML is the foundation for the Web Services Description Language (WSDL). (http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci750567,00.html).

Regardless of this, however, Applicant has expressly provided its definition of Web services and Applicant's express definition of Web services cannot legitimately be read on all software as long as it is used in connection with a network. The entire specification makes that clear beyond any reasonable doubt.

All of the definitions quoted above are basically consistent with Applicant's definition. Yet the Office refuses to accept this definition even though it is both the ordinary and usual meaning as well as Applicant's definition as provided in the specification.

Rather, the Office, seemingly randomly, defines "Web services" as any software appearing on a network, which certainly is not the ordinary and usual meaning of that term or Applicant's definition. Certainly, it is not Zintel's definition. Zintel does not use the term "Web services" at all. Zintel simply refers to "Services" and expressly defines that term in column 9, lines 1-12 as follows:

Service. The fundamental UPnP control entity (but not the finest level of control). An example of a Service is "Clock". Services are defined with a mandatory common base set of functionality. Vendors can extend base functionality with proprietary extensions provided the base functionality is implemented. Service definitions are versioned and later versions are

constrained to be supersets of previous versions. UPnP enables searches for all Devices that contain a specified Service of a minimum version. This search would find all clocks, regardless of their packaging. The search for Device Type "Clock" would be used to find only stand-alone clocks.

Accordingly, it is quite clear that Zintel's "Services" have nothing to do with the present invention's "Web services".

The Office is reading "Web service" to cover at least any software on a node of a network that can be accessed and used by another node on the network.

This definition is not warranted by the claim language, especially the amended claim language. Specifically, the language of reconfiguring the Web services available at a node used in claim 1, for instance, cannot reasonably be read on merely exchanging information about software. It inherently means configuring the software itself and not just accessing and using it. Applicant has, nevertheless, amended independent claims 1 and 25 to expressly recite that the reconfiguring includes exchanging software modules between containers. As such, there should no longer be any dispute that claims 1 and 25 do not read on Zintel since Zintel discusses only exchanging information about "Services", and not the Services themselves.

The UPnP networking and SSDP disclosed in columns 57 and 58 of Zintel referred to by the Office does not concern the exchanging of software modules. Rather, it simply refers to the exchange of information about software. Accordingly, claims 1 and 25 patentably distinguish over Zintel for the reasons set forth above and in the response to the previous Office Action.

With respect to dependent claims 2 and 26, the distinctions are even more clear. Claim 2 depends from claim 1 and adds that the instructions for causing the dynamic reconfiguration of Web services comprises instructions for transmitting messages "requesting said other containers to return copies of Web services software" and instructions, responsive to receipt of those requests "for sending copies of said requested Web services software to sit requesting containers". The Office asserts that this claim language reads on Zintel's disclosure of sending messages requesting the UPnP description of a device's services and their parameters and controls and the

return of such information to the requester reads on this language. However, it clearly does not. Requesting and receiving information about a software module is totally different than requesting and receiving the module itself.

The Office attempts to get around this by asserting that "the XML syntax of the description is Web services software". Accordingly, with respect to claim 2, the Office is going even further than it did in connection with claim 1 and asserting that Web services, not only includes any software available on a network that can be accessed and used by another node, but also encompasses simple XML syntax contained in a message between two nodes. This definition is particularly inappropriate as it seems to (1) have no connection whatsoever to any definition in Zintel of "Service" (2) have no rational relationship to the ordinary and usual meaning of the word "service", and (3) have no rational relationship to Applicant's express definition or use of the term "Web services" in the specification.

The Office is attempting to place Applicant in the impossible position of being denied the ability (1) to use the ordinary and usual meaning of "Web services" (or even just the word "service") or to define the term itself. This leaves Applicant with no option for claiming what Applicant would like to claim. The Office is reading "Web services" as covering XML syntax inside of a network message. Thus, the Office is reading "Web services" as meaning "code appearing on a network". Certainly, when stated this way, it should be apparent that this is not a justifiable interpretation of the claim language, even if one were to ignore the specification and read the terminology in the abstract.

Claim 26 depends from claim 25 and is similar in scope to claim 2. Accordingly, these two claims even further distinguish over the prior art of record.

Dependent claims 4 and 28 recite the feature whereby the containers perform their discovery by querying a Web services registry. The Office asserted that Zintel's contacting of the UPnP template to find out the UPnP description for a device as discussed in column 58, lines 35-65 meets this limitation.

Again, the Office simply is not giving the claim term "web services registry" a rational interpretation. Zintel's UPnP templates are associated with the particular node

to which they correspond. In fact, as near as Applicant understands the prior art rejections in this case, the UPnP template appears to be what the Office considers (erroneously) to correspond to the claimed "containers". Thus, the Office is recharacterizing the very same template that it needs to rely on as constituting the "container" as a Web services registry. This is especially improper here where Applicant has bent over backwards to make clear that this claim recites an alternative to discovery via querying containers. Specifically, the specification discusses four different ways to discover Web services information, including (1) peer-to-peer inquiry, i.e., two containers directly communicating with each other, and (2) querying of a Web services registry. See page 11, line 8 through page 12, line 3. Thus, Applicant's specification clearly describes that the claimed Web services registry is not the claimed container.

Even further, Web services registries are well known in the field and no one skilled in these arts would consider Zintel's template to be a Web services registry. If this is not enough, Applicant has clearly set forth what it means by "Web service registry" by referring to specific examples of available Web services directories. See page 4, lines 4-11 of the specification, where it states:

The UDDI initiative is an XML-based registry standard by which businesses list themselves and the Web services they offer on the Internet. WSDL is one approach to describing such Web services. A key goal of the UDDI initiative is to enable companies to find each other and each other's Web services on the Internet and to make their computer systems inter-operable with each other in order to facilitate electronic commerce. The UDDI initiative allows businesses to list information about themselves, e.g., name, location, and/or the Web services they offer.

There is no rational interpretation of the term "Web services registry" that covers Zintel's template, especially when the Office already is using the template as the very "container" that this claim language is intended to distinguish from. Thus, this rejection is improper and should be withdrawn.

Dependent claims 5 and 29 recite that the messages are in WSDL. The Office asserted that this is taught in secondary reference Christensen and that it is obvious to combine Christensen with Zintel because Zintel uses XML to describe network services and Christensen teaches a format that is a known XML format variant for describing network services.

This is an improper combination of references. WSDL is an acronym for Web Services Descriptor Language. WSDL is a language developed for use in connection with Web services. The concept of using WSDL in connection with Zintel's registering of nodes on the network, which has nothing to do with Web services, makes no sense. WSDL is designed for an entirely different purpose.

Furthermore, the Office rejected dependent claims 6 and 30 as being unpatentable over Zintel, Christensen and Januszewski, noting that Januszewski discusses UDDI. However, once again, the combination is improper and this rejection further highlights the fact that the Office is over broadly interpreting Applicant's claim language. First of all, Januszewski is an unnecessary reference since the Office has cited Januszewski solely to show that UDDI is a known Web services registry and Applicant admits that UDDI is a known Web services registry. Nevertheless, the proposed combination is improper because, in this combination, the Office is saying that Zintel's template can be the UDDI registry. The UDDI registry is a web site (www.uddi.org) at which businesses register their Web services and make them available to others. The assertion that Zintel's template is the website www.uddi.org is pure nonsense.

Conclusion

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: June 26, 2006

/Theodore Naccarella/
Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

TXN:pmf